Japanese Patent Laid-Open No.7-28736

(54) [Title of the Invention]

MULTIWINDOW CONTROL SYSTEM

(57) [Abstract]

The present invention relates to a multiwindow control mechanism and its object is to reduce the quantity of data referred to as events or requests to be transferred between a client 5 and a server 6.

[Constitution]

Event management means 63 for designating that the processing corresponding to a generated event is carried out in the server 6 and script management means 62 for entering a script in which the content of the processing is described are provided to execute a previously entered script in the server 6 when an event is generated.

[Advantage]

When the client 5 and the server 6 are operated by different computers connected by a network, the quantity of the data to be transferred between the client 5 and the server 6 when an even is generated is decreased and thereby, the communication performance of the network is not deteriorated.

[Claims]

[Claim 1]

A multiwindow control system using a multiwindow control mechanism constituted of an application program referred to as client and a server to be operated on a computer provided with an input unit, an output unit, and a storage area to display a character and a pattern on the output unit and having a function for supplying the data

referred to as request in which a display request for a character and a pattern to the output unit is described to the server from the client and a function for supplying the data referred to as event in which operational contents of the input unit are described to the client from the server,

characterized by comprising:

drawing means for outputting an instruction for making the output unit display a character and a pattern in accordance with a request sent from the client, the data referred to as script in which processing contents of an indication or the like to be executed by a server when an event is generated in accordance with a combination of a series of requests are described;

script management means for storing the script of the entry request for received data, generating the script control information in which the type of the script and the storing place are described, decoding a designated script when a script execution request is generated, and sending the request described in the script to the drawing means;

event management means for generating the event control information in which the type of an event to be processed and type of a script to be executed when the event is generated are described when receiving an entry request for an event for a client to designate the processing destination of the event, generating an event when the input unit is operated, and transferring an event to a client in accordance with the event control information or issuing a script execution request to the script management means; and

module management means for relaying an event and a request between a client, the drawing means, the scrip management means, and the event management means.

[Claim 2]

The multiwindow control means according to claim 1, characterized in that means for distributing the processing of a generated event to the client side or server side is included.

[Detailed Description of the Invention]

[0001]

[Field of the invention]

The present invention relates to the processing of software generated by operating an input unit such as a mouse or keyboard of a computer.

[0002]

[Description of the prior art]

A conventional multiwindow control system uses a mechanism shown in Fig. 2.

[0003]

In Fig. 2, symbol 1 denotes a body of a computer, 2 denotes a network for performing communication between computers, 3 denotes an output unit such as a display, 4 denotes an input unit such as a mouse or keyboard. Symbol 5 denotes an application program referred to as client. Symbol 6 denotes a server for performing a service corresponding to a request of the client 5, which has a function for displaying a character and a pattern on a display when generating the data (hereafter referred to as event) in which a position of a mouse after moved is recorded and sending the data to the client 5 or the data (hereafter referred to as request) in which a request

for displaying a character or pattern on a display is supplied from the client 5. It is permitted to operate the client 5 and sever 6 on the same computer or different computers. At the time of operating the client 6 and server 6 on different computers, an event or request is exchanged through communication means such as the network 2.

[0004]

In the case of a multiwindow control system having the above configuration, when a user of the computer 1 applies any operation to the input unit 4, a change of states of the input unit 4 is communicated to the server 6, and the server 6 generates an event and transfers the event to the client 5. The client 5 decodes the event and outputs a request corresponding to the decoded event to the server 6. Moreover, when the request is a display request, the server 6 makes the output unit 3 display a character or pattern requested by the client.

[0005]

Moreover, there is the following system based on the configuration in Fig. 2.

[0006]

For example, when a user of the computer 1 moves a mouse and then displays the movement trace of the mouse, an event to be communicated to the client 5 after movement of the mouse is started is an even showing a momentary position mouse position. Thus, when types of a series of events previously generated are changed, the multiwindow control system disclosed in the official gazette of Japanese Patent Laid-Open No. 283622/1989 uses a method that a processing part of the above events is previously set in a server 6, the information for making the server 6 perform a processing corresponding to an event

and the control data in which an ending condition of the processing are described are sent to the server 6 from the client 5 but the server 6 does not send generated events to the client 5 until the ending condition is satisfied, and the events are processed by the processing part of the server 6 in accordance with the control data.

[0007]

[Problem to be solved by the invention]

In the case of the above multiwindow control method, however, when a user frequently operates the input unit 4 while the client 5 and server 6 are operated by different computers 1, the numbers of events and requests to be exchanged between the client 5 and server 6 are increased. Therefore, the quantity of the data circulating through the network 2 is increased to lower the communication rate of another computer 1 using the network 2.

[0008]

Moreover, in the case of Japanese Patent Laid-Open No. 1-283662, it is necessary for the server 6 to have a processing part corresponding to all control data values sent from the client 5 and therefore, a program of the server 6 becomes gigantic. Therefore, problems occur that the development cost of the server 6 increases and maintenance becomes difficult.

[0009]

[Means for solving the problems]

A multiwindow control system of the present invention is characterized by including: drawing means for outputting an instruction for displaying a character and a pattern on an output unit in accordance with a request sent from a client, the data (hereafter referred to as script) describing the processing content

of an indication or the like to be executed by a server when an event is generated in accordance with a combination of a series of requests; script management means for storing the script when receiving the entry request of the script, generating the script control information in which the type and storing place of the script are described, decoding a designated script when a script execution request is generated, and sending the request described in the script to the drawing means; event management means for generating the event control information in which the type of an event to be processed when receiving an event entry request for a client to designate the processing destination of the event and the type of a script to be executed when the event is generated are described, generating an event when an input unit is operated, and transferring an event to a client or issuing a script execution request to the script management means in accordance with the event control information; and module management means for relaying events and requests between a client, the drawing means, the script management means, and the event management means.

[0010]

[Function]

Module management means relays events and requests between a client, script management means, event management means, and drawing means. When a request is an event entry request, the event management means generates the event control information corresponding to the request. When a request is a script entry request, the script management means stores the script and generates the script control information corresponding to the script. When the request is a request other than the script entry request, the drawing means

performs the processing such as indication corresponding to the request.

[0011]

When a state change of an input unit is communicated to event management means through an operation of the input unit by a computer user, the event management means generates an event.

[0012]

When the event control information for the event is present and it is designated for a client to process the event, it is possible to execute a corresponding processing by sending the event to the client.

[0013]

Moreover, when the event control information for the event is present in the event management means and the type of a script is designated to the event control information, the event management means requests script management means to execute the script and the script management means decodes the designated script and supplies a series of requests described in the script to drawing means, and the drawing means performs the processing such as indication corresponding to the request similarly to the case of a request from a client. Therefore, it is possible to process the correspondence to an event only in a server and reduce the numbers of requests and events to be transferred between the client and the server.

[0014]

[Embodiment]

Fig. 1 shows a configuration of a multiwindow control system of an embodiment of the present invention.

[0015]

In Fig. 1, symbol 3 denotes an output unit such as a display, 4 denotes an input unit such as a mouse or keyboard, and 5 denotes an application program referred to as a client. Symbol 6 denotes a server for performing a service corresponding to a request of the client 5, which has a function for generating an event in which a position of a mouse after moved and sending it to the client 5 or displaying a character and a pattern on the output unit 3 when there is sent a request for displaying a character and a pattern from the client 5. It is permitted to operate the client 5 and sever 6 on the same computer or different computers. At the time of operating them on different computers, events and requests are exchanged through communication means such as a network. Symbol 61 denotes drawing means for outputting an instruction for making the output unit 3 display a character and a pattern in accordance with a request sent from the client 5 or the like. Symbol 62 denotes script management means for entering a script to be executed when an event is generated, decoding a designated script when a script execution request is generated, and outputting the execution request of a request described in a script to the drawing means 61 in accordance with client 5. Symbol 63 denotes event management means for entering the processing destination of an event when generated, generating an event when the input unit 4 is operated, and transferring the event to the client 5 or outputting a script execution request to the script management means 62 in accordance with client 5. Symbol 64 denotes module management means for relaying events and requests between the client 5, drawing means 61, script management means 62, and event management means 63. Symbol 65 denotes a request storage area serving as a

constant area in a memory area of a computer to store requests. Symbol 66 denotes an event storage area serving as a constant area in a memory area of a computer to store events. In the case of this embodiment, the drawing means 61, script management means 62, and event management means 63 are respectively realized by software. However, it is also possible to realize them by hardware.

[0016]

Fig. 3 shows an example of the event control information generated by the event management means 63, which is realized as the information recorded in a memory area of a computer. In Fig. 3, symbol 631 denotes an area in which the identification code of the client 5 requesting entry processing of an event is recorded. Symbol 632 denotes an area in which the identification code of an event is recorded. The identification code is uniquely assigned by the event management means 63 when the input unit 4 is operated and it is assumed that the client 5 knows the code. Symbol 633 denotes an area in which the identification code of a script to be executed when processing a generated event in the server 6 is recorded. The code is the same as a code to be uniquely assigned to a script when the client 5 requests entry of the script.

[0017]

Fig. 4 shows an example of the script information generated by the script management means 62, which is realized as the information recorded in a memory area of a computer. In Fig. 4, symbol 621 denotes an area in which the identification code of the client 5 requesting entry processing of a script is recorded. Symbol 622 denotes an area in which the identification code of a script entry-requested by the client 5 is recorded. The code is uniquely assigned by the client

5. Symbol 623 denotes an area in which a script entry-requested is recorded in a computer and then, the place of the memory area is recorded.

[0018]

Fig. 5 is a flowchart showing operations of the module management means 64.

[0019]

Fig. 6 is a flowchart showing operations of the event management means 63.

[0020]

Fig. 7 is a flowchart showing operations of the script management means 62.

[0021]

Fig. 8 is a flowchart showing operations of the drawing means 61.

[0022]

Operations of the multiwindow control system of this embodiment constituted as described above will be described below.

[0023]

First, the module management means 64 checks if a request is sent from the client 5 through request determination 641. When no request is sent, the means 64 executes input determination 648. When a request is sent, the means 64 records the request in the request storage area 65 through request storing processing 642. It is assumed that a request is constituted of the identification code of the client 5 sending the request, identification code of the request, and additional information intrinsic to each request. After recording

the request, the type of the request is checked through event entry request determination 643 or script entry request determination 645. [0024]

In the case of a request for requesting entry of an event, execution is shifted to the event management means 63 by event management processing 644. It is assumed that the identification code of an event to be generated by operating the input unit 4, a code showing whether to perform the processing corresponding to the event by the client 5 or server 6, and the identification code of a script to be executed when the processing corresponding to the event is performed by the server 6 are added to the request. The event management means 63 first checks whether a request is present through request determination 634. In this case, because a request is present, the event control information shown in Fig. 3 is generated in a memory area of a computer through event entry 635, the identification code of the client 5 added to the request is recorded in the client part 631 of the area, and the identification code of the event added to the request is recorded in an event part 632. In the case of a request for requesting that the processing corresponding to an event is performed by the client 5, no data is recorded in the script part 633. In the case of a request for requesting that the processing corresponding to the request is performed by the server 6, the identification code of a script added to the request is recorded in the script part 633. As described above, an event is entered and then, the processing by the event management means 63 is completed, and execution is shifted to input determination 648 of the module management means 64.

[0025]

In the case of a request for requesting entry of a script, execution is shifted to the script management means 62 through script management processing 646. In this case, it is assumed that a script to be entered and the identification code of the script are added to the above request. The script management means 62 first checks presence or absence of a request through the request determination 624. In this case, because a request is present, a script is recorded in a memory area of a computer through the script entry 625. Moreover, the script control information shown in Fig. 4 is generated in another memory area of the computer, the identification code of the client 5 added to the request is recorded in the client part 621 of the area, the identification code of the script added to the request is recorded in the script part 622, and the place of the memory area in which the script is recorded is recorded in a storage part 623. As described above, after entering the script, the processing by the script management means 62 is completed and execution is shifted to the input determination 648 of the module management means 64.

[0026]

When a request does not request entry of an event nor entry of a script, execution is shifted to the drawing means 61 through drawing processing 647. The drawing means 61 determines various types of requests and performs the processing corresponding to a request. For example, a request for requesting character drawing or not is determined through character drawing determination 611. In the case of a request for requesting character drawing, the output unit 3 is made to display a character through character drawing 612 and

execution is shifted to the input determination 648 by the module management means 64 again.

[0027]

The module management means 64 during the operation of the server 6 regularly determines whether the input unit 4 is operated by a user of a computer through the input determination 648. When the input unit 4 is not operated, the request determination 641 is restarted because it is unnecessary to generate an event. When the input unit 4 is operated, it is determined through end operation determination 649 whether operation is the end operation of the server 6. When the operation is the end operation, execution of the server 6 is completed. When the operation is not the end operation, execution is shifted to the event management means 63 through the event management 644.

[0028]

The event management means 63 first checks presence or absence of a request through the request determination 634. Because request processing is already completed, the means 63 generates an event in accordance with the operation content of the input unit 4 or the like through the next event generation 636, assigns the identification code of the event, and records the code in the event storage area 66 depending on the operational contents. Then, the means 63 determines presence or absence of an event through event entry determination 637. When the event control information generated for the client 5 to be operated by the input unit 4 is present and a code same as the identification code of the generated event is recorded in the event part 632, the means 63 executes processing determination 638 by assuming that an even is recorded. When the event control information is absent or a code same as the identification code of

the event generated in the event part 632 of the event control information is not recorded, the means 63 completes the processing of the event management means 63 without doing anything by assuming that it is unnecessary to perform the processing corresponding to an event and execution is shifted to event transfer determination 6410 of the module management means 64. The processing determination 638 determines whether to perform the processing corresponding to the event by the server 6 or client 5. When no data is recorded in the script part 633 of event control information, by assuming that the processing corresponding to an event is performed by the client 6, event transfer request 639 requests the module management means 64 to transfer an event to the client 6 and completes the processing of the event management means 63, and execution is shifted to the event transfer determination 6410 of the module management means 64. When the identification code of a script is recorded in the script part 633 of the event control information, by assuming that the processing corresponding to an event is performed by the server 6, a script execution request 6310 requests the execution request of the script shown by the identification code to the script management means 62 and completes the processing of the event management means 63, and execution is shifted to the event transfer determination 640 of the module management means 64.

[0029]

The module management means 64 determines presence or absence of an event transfer request to the client 6 through the event transfer determination 6410. When an event transfer request is generated, the means 64 reads an event from the event storage area 66 through event

transfer processing 6411 and transfers the event to the client 5, and the request determination 641 is restarted.

[0030]

When an event transfer request is not generated, it is determined through script execution determination 6412 whether a script execution request is generated from the event management means 63 to the script management means 62. When a script execution request is not generated, the request determination 641 is restarted. When a script execution request is generated, execution is shifted to the script management means 62 through the script management processing 646.

[0031]

The script management means 62 first checks presence or absence of a request through the request determination 624. Because the processing of a request is already completed, the means 62 executes a script requested from the event management means 63 through the next script execution 626. When a request to be executed by the drawing means 61 is included in the script, the means 62 records the request in the request storage area 65, outputs a drawing request to the drawing means 61, and completes the processing of the script management means 62, and execution is shifted to drawing processing determination 6413 of the module management means 64.

[0032]

The module management means 64 determines through the drawing processing determination 6413 whether a drawing request is generated from the script management means 62 to the drawing means 61. When a drawing request is not generated, the request determination 641

is restarted. When a drawing request is generated, execution is shifted to the drawing means 61 through the drawing 647.

[0033]

The drawing means 61 performs the processing corresponding to a request recorded in the request storage area 65 and then completes the processing of the drawing means 61, and execution is shifted to the request determination 641 of the module management means 64.

[0034]

[Advantage of the invention]

As described above, in the case of the present invention, even when a user of a computer frequently operates an input unit, the processing time for exchanging events and requests between a client and a server does not increase because the numbers of the events and requests are small or execution speeds of the client and server are not lowered. Moreover, when the client and the server operate on different computers, the data quantity circulating through a network does not increase or the communication rate of other computer using the same network is not lowered.

[0035]

Furthermore, because the present invention performs the processing corresponding to an event which is originally performed by a server by a script supplied from a client, the program of the server does not become gigantic and thereby, the sever development cost decreases and maintenance becomes easy.

[Brief Description of the Drawings]

Fig. 1 is a block diagram showing an embodiment of the present invention;

Fig. 2 is a block diagram showing an embodiment of the present invention and a conventional example;

Fig. 3 is an illustration showing event control information:

Fig. 4 is an illustration showing script control information;

Fig. 5 is a flowchart showing operations of module management means;

Fig. 6 is a flowchart showing operations of event management means;

Fig. 7 is a flowchart showing operations of script management means; and

Fig. 8 is a flowchart showing operations of drawing means.

[Description of symbols]

1 ... Computer

2 ... Network

3 ... Output unit

4 ... Input unit

5 ... Client

6 ... Server

61 ... Drawing means

62 ... Script management means

63 ... Event management means

64 ... Module management means

65 ... Request storage area

66 ... Event storage area

622 ... Script part

623 ... Storage part

631 ... Client part

632 ... Event part

633 ... Script part

Fig. 1

3 Output unit

4 Input unit

5 Client

6 Server

61 Drawing means

62 Script management means

63 Event management means

64 Module management means

65 Request storage area

66 Event storage area

Fig. 2

1 Computer

2 Network

3 Output unit

4 Input unit

5 Client

6 Server

Fig. 3

631 Client part

632 Event part

633 Script part

Fig. 4

621 Client part

622 Script part

623 Storage part

Fig. 5

641 Request determination

642 Request storage processing

643 Event entry request determination

644 Event management processing

645 Script entry request determination

646 Script management processing

647 Drawing processing

648 Input determination

649 End operation determination

6410 Event transfer determination

6411 Event transfer processing

6412 Script execution determination

6413 Drawing determination

Fig. 6

634 Request determination

635 Event entry

636 Event generation

637 Event entry determination

638 Processing determination

639 Event transfer request

6310 Script execution request

#1 Script execution

#2 Event transfer

Fig. 7

624 Request determination

625 Script entry

626 Script execution

Fig. 8

611 Character drawing determination

612 Character drawing processing

613 Circle drawing determination

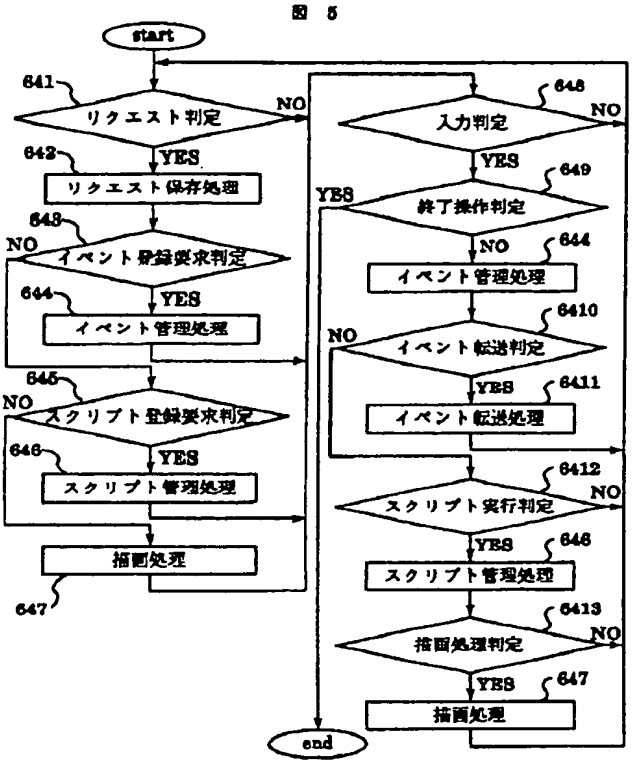
614 Circle drawing processing

615 Color change determination

616 Color change processing

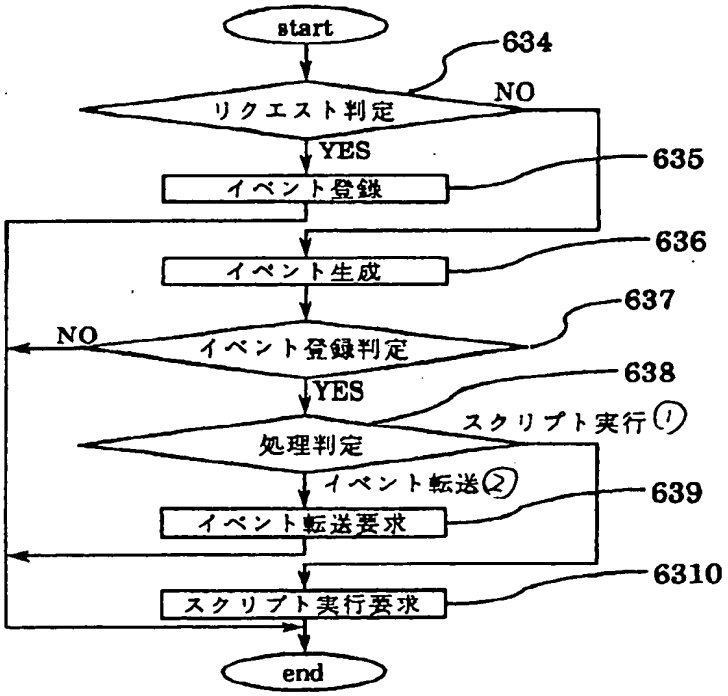
【図５】

図 5


start
641
リクエスト判定
NO
YES
642
リクエスト保存処理
643
イベント登録要求判定
NO
YES
644
イベント管理処理
645
スクリプト登録要求判定
NO
YES
646
スクリプト管理処理
描画処理
647
648
入力判定
NO
YES
649
終了操作判定
YES
NO
644
イベント管理処理
6410
イベント転送判定
NO
YES
6411
イベント転送処理
6412
スクリプト実行判定
NO
YES
646
スクリプト管理処理
6413
描画処理判定
NO
YES
647
描画処理
end

【図６】

図　6


start
634
リクエスト判定
NO
YES
635
イベント登録
636
イベント生成
637
イベント登録判定
NO
YES
638
処理判定
スクリプト実行①
イベント転送②
639
イベント転送要求
6310
スクリプト実行要求
end

(19)日本国特許庁(JP)　(12)公開特許公報（A）　(11)特許出願公開番号

特開平７－２８７３６

(43)公開日　平成7年(1995)1月31日

(51)Int. Cl.[6]		識別記号	庁内整理番号	F I	技術表示箇所
Ｇ０６Ｆ	13/00	３５７　Ｚ	7368－５Ｂ		
	3/14	３５０　Ａ			

審査請求　未請求　請求項の数２　ＯＬ　（全８頁）

(21)出願番号　特願平5-175104

(22)出願日　平成5年(1993)7月15日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地

(71)出願人　000153476
株式会社日立マイクロソフトウェアシステムズ
神奈川県横浜市戸塚区吉田町292番地

(72)発明者　米澤　知江
神奈川県横浜市戸塚区吉田町292番地株式会社日立製作所マイクロエレクトロニクス機器開発研究所内

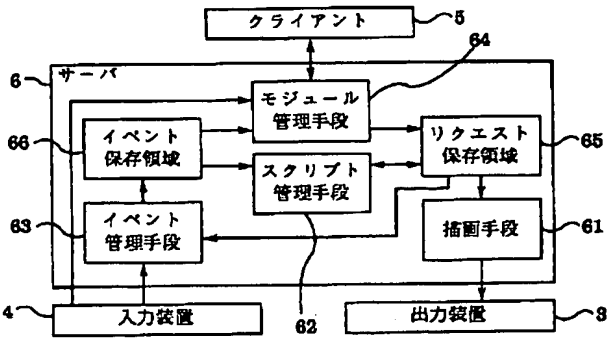
(74)代理人　弁理士　小川　勝男

最終頁に続く

(54)【発明の名称】マルチウィンドウ制御方式

(57)【要約】
【目的】本発明はマルチウインドウ制御機構に関し、その目的は、クライアント５、サーバ６間でやり取りするイベントやリクエストと呼ばれるデータの量を低減することにある。
【構成】イベント発生時の対応処理をサーバ６内部で行なうことを指定するためのイベント管理手段６３と、その対応処理内容を記述したスクリプトを登録するためのスクリプト管理手段６２を設け、イベント発生時に予め登録されているスクリプトをサーバ６内で実行する。
【効果】クライアント５とサーバ６がネットワークで繋がれた別々のコンピュータで動作する場合に、イベント発生時のクライアント５とサーバ６間でやり取りするデータ量が減り、同ネットワークの通信性能の低下を招くことが無くなる。

図 1



1

【特許請求の範囲】

【請求項１】クライアントと呼ばれるアプリケーションプログラムと、入力装置、出力装置、記憶領域を有するコンピュータ上で動作し、出力装置へ文字や図形の表示等を行うサーバから成り、リクエストと呼ばれる、出力装置への文字や図形の表示要求等を記述したデータを、クライアントからサーバに渡す機能と、イベントと呼ばれる、入力装置の操作内容等を記述したデータを、サーバからクライアントに渡す機能とを有するマルチウインドウ制御機構において、クライアントから送られてくるリクエストに応じて、出力装置に文字や図形を表示させるための命令を出す描画手段と、スクリプトと呼ばれる、イベント発生時にサーバで実行する表示等の処理内容を一連のリクエストの組合せで記述したデータと、その登録要求を受け取った時に、そのスクリプトを保存し、スクリプト種別と保存場所等を記述したスクリプト制御情報を作成し、スクリプトの実行要求が発生した場合に指定されたスクリプトを解読し、スクリプトに記述されているリクエストを前記描画手段に送るスクリプト管理手段と、イベントの処理先等をクライアントが指示するためのイベントの登録要求を受け取った場合に、処理を行なうべきイベントの種別や、そのイベントが発生した場合に実行するスクリプトの種別等を記述したイベント制御情報を作成し、入力装置の操作等が行なわれた場合にイベントを生成し、イベント制御情報に基づいて、クライアントへのイベント転送、または前記スクリプト管理手段へのスクリプト実行要求の発行等を行なうイベント管理手段と、クライアント、前記描画手段、前記スクリプト管理手段、前記イベント管理手段との間でイベント及びリクエストの中継を行なうモジュール管理手段、とを備えたことを特徴とするマルチウインドウ制御方式。

【請求項２】請求項１において、イベント発生時にその処理をクライアント側またはサーバ側に分配する手段を備えたことを特徴とするマルチウインドウ制御方式。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、マウスやキーボード等の入力装置を持つコンピュータに於いて、それらの入力装置の操作により発生するソフトウエアの処理に関する。

【０００２】

【従来の技術】従来のマルチウインドウ制御方式は、図２に示すような機構を用いていた。

【０００３】図２において１はコンピュータの本体、２はコンピュータ間で通信を行なうためのネットワーク、３はディスプレイなどの出力装置、４はマウスやキーボードなどの入力装置である。５はクライアントと呼ばれるアプリケーションプログラムである。６はクライアント５の要求に応じたサービスを行なうサーバであり、例

2

えば、マウスを動かした場合などに、マウスの移動後の位置などを記録したデータ（以降ではイベントと呼ぶ）を生成してクライアント５に送ったり、クライアント５から、ディスプレイへの文字や図形の表示要求等を記述したデータ（以降ではリクエストと呼ぶ）が送られてきた場合などに、ディスプレイに文字や図形を表示させる機能を持つ。クライアント５とサーバ６は同じコンピュータ上でも、あるいは別々のコンピュータ上でも動作して良い。別々のコンピュータ上で動作する場合は、ネットワーク２などの通信手段を通じてイベントやリクエストのやり取りを行なう。

【０００４】このような構成のマルチウインドウ制御方式においては、コンピュータ１の利用者が入力装置４に何らかの操作をすると、サーバ６に入力装置４の状態の変化が通知され、サーバ６はイベントを生成しクライアント５へ転送する。クライアント５はイベントを解読し、それに応じたリクエストをサーバ６に出す。さらにそのリクエストが表示要求である場合、サーバ６はクライアントの要求する文字や図形を出力装置３に表示させていた。

【０００５】また図２の機構を基にした次のような方式がある。

【０００６】例えばコンピュータ１の利用者がマウスを移動させたときにその移動軌跡を表示するような場合、マウスの移動が始まった後クライアント５に通知されるイベントは、時々刻々のマウスの位置を示すイベントである。このように予め発生する一連のイベントの種類がわかる場合に、特開平１－２８３６２２号公報のマルチウインドウ制御方式では、そのようなイベントの処理部を予めサーバ６内部で持ち、イベントに応じた処理をサーバ６側で行なわせるための情報と、その処理の終了条件とを記述した制御データを、クライアント５からサーバ６に送ると、サーバ６は終了条件を満たすまでの間、発生したイベントをクライアント５に転送せずに、その制御データを基にサーバ６の処理部により処理するという方法をとっていた。

【０００７】

【発明が解決しようとする課題】しかしながら上記のようなマルチウインドウ制御方式においては、クライアント５とサーバ６が別々のコンピュータ１で動作している場合に、利用者が頻繁に入力装置４を操作すると、クライアント５とサーバ６の間でやり取りするイベント、及びリクエストの量が増えることになる。このため、ネットワーク２に流れるデータ量が増え、そのネットワーク２を利用している他のコンピュータ１の通信速度を低下させることになる。

【０００８】また特開平１－２８３６２２号公報の場合、サーバ６はクライアント５から送られてくる総ての制御データに対応する処理部を持つ必要があり、サーバ６のプログラムが巨大になってしまう。そのためサーバ

６の開発コストが高くなり、また保守が困難になるという問題があった。

【０００９】

【課題を解決するための手段】本発明のマルチウインドウ制御方式は、クライアントから送られてくるリクエストに応じて、出力装置に文字や図形を表示させるための命令を出す描画手段と、イベント発生時にサーバで実行する表示等の処理内容を一連のリクエストの組合せで記述したデータ（以降ではスクリプトと呼ぶ）と、その登録要求を受け取った場合に、そのスクリプトを保存し、スクリプト種別と保存場所等を記述したスクリプト制御情報を作成し、スクリプトの実行要求が発生した場合に指定されたスクリプトを解読し、スクリプトに記述されているリクエストを前記描画手段に送るスクリプト管理手段と、イベントの処理先等をクライアントが指示するためのイベントの登録要求を受け取った場合に、処理を行なうべきイベントの種別や、そのイベントが発生した場合に実行するスクリプトの種別等を記述したイベント制御情報を作成し、入力装置の操作等が行なわれた場合にイベントを生成し、イベント制御情報に基づいて、クライアントへのイベント転送、または前記スクリプト管理手段へのスクリプト実行要求の発行等を行なうイベント管理手段と、クライアント、前記描画手段、前記スクリプト管理手段、前記イベント管理手段との間でイベント及びリクエストの中継を行なうモジュール管理手段、とを備えたことを特徴とする。

【００１０】

【作用】モジュール管理手段は、クライアント、スクリプト管理手段、イベント管理手段、描画手段との間でイベント及びリクエストの中継を行なう。リクエストがイベントの登録要求である場合はイベント管理手段がその要求に応じたイベント制御情報を作成し、リクエストがスクリプトの登録要求である場合はスクリプト管理手段がスクリプトを保存しそのスクリプトに応じたスクリプト制御情報を作成し、それ以外のリクエストである場合は描画手段がそのリクエストに対応する表示等の処理を行なう。

【００１１】コンピュータ利用者による入力装置の操作等で、イベント管理手段に入力装置の状態変化が通知されると、イベント管理手段はイベントを生成する。

【００１２】そのイベント用のイベント制御情報が存在し、且つイベントの処理をクライアントで行なうということが指示されている場合は、そのイベントをクライアントに送り、対応する処理を実行することが出来る。

【００１３】また、イベント管理手段にそのイベント用のイベント制御情報が存在し、且つイベント制御情報にスクリプトの種類が指定されている場合は、スクリプト管理手段にそのスクリプトの実行を要求し、スクリプト管理手段は指定されたスクリプトを解読し、スクリプト中に記述されている一連のリクエストを描画手段に渡し、描画手段はクライアントからのリクエストと同じように、そのリクエストに対応する表示等の処理を行なう。このためイベントへの対応をサーバ内部のみで処理することができ、クライアントとサーバ間でやり取りするリクエストやイベントの量を減らすことが出来る。

【００１４】

【実施例】図１は本発明の一実施例におけるマルチウインドウ制御方式の構成を示すものである。

【００１５】図１において３はディスプレイなどの出力装置、４はマウスやキーボードなどの入力装置である。５はクライアントと呼ばれるアプリケーションプログラムである。６はクライアント５の要求に応じたサービスを行なうサーバであり、例えば、マウスを動かした場合などに、マウスの移動後の位置などを記録したイベントを生成してクライアント５に送ったり、クライアント５から文字や図形の表示要求等を記述したリクエストが送られてきた場合などに、出力装置３に文字や図形を表示させる機能を持つ。クライアント５とサーバ６は同じコンピュータ上でも、あるいは別々のコンピュータ上でも動作して良い。別々のコンピュータ上で動作する場合は、ネットワークなどの通信手段を通じてイベントやリクエストのやり取りを行なう。６１はクライアント５などから送られてくるリクエストに応じて、出力装置３に文字や図形を表示させるための命令を出す描画手段である。６２はクライアント５の要求に応じて、イベント発生時に実行するスクリプトを登録し、スクリプトの実行要求が発生した場合に指定されたスクリプトを解読し、描画手段６１にスクリプトに記述されたリクエストの実行要求を出すスクリプト管理手段である。６３はクライアント５の要求に応じて、イベント発生時のイベントの処理先等を登録し、入力装置４の操作等が行なわれた場合にイベントを生成し、クライアント５へのイベント転送、またはスクリプト管理手段６２へのスクリプト実行要求を出すイベント管理手段である。６４はクライアント５と、描画手段６１、スクリプト管理手段６２、イベント管理手段６３との間でイベント及びリクエストの中継を行なうモジュール管理手段である。６５はリクエストを保存するための、コンピュータの記憶領域中の一定領域であるリクエスト保存領域である。６６はイベントを保存するための、コンピュータの記憶領域中の一定領域であるイベント保存領域である。なお、本実施例においては、描画手段６１、スクリプト管理手段６２、イベント管理手段６３はいずれもソフトウエアで実現するものとするが、ハードウエアで実現することも可能である。

【００１６】図３はイベント管理手段６３が作成するイベント制御情報の一例を示し、コンピュータの記憶領域に記録された情報として実現する。図３において６３１はイベントの登録処理を要求しているクライアント５の識別記号を記録する領域である。６３２はイベントの識

別記号を記録する領域であり、この記号は入力装置４の操作等が行なわれたときにイベント管理手段６３で一意に割り当てられる記号であり、クライアント５はその記号を知っているものとする。６３３はイベント発生時にその処理をサーバ６内で行なう場合に実行するスクリプトの識別記号を記録する領域である。この記号はクライアント５がスクリプト登録要求時に、スクリプトに一意に割り当てる記号と同じである。

【００１７】図４はスクリプト管理手段６２が作成するスクリプト制御情報の一例を示し、コンピュータの記憶領域に記録された情報として実現する。図４において６２１はスクリプトの登録処理を要求しているクライアント５の識別記号を記録する領域である。６２２はクライアント５から登録要求のあったスクリプトの識別記号を記録する領域である。この記号はクライアント５が一意に割り当てる記号である。６２３は登録要求のあったスクリプトをコンピュータの記憶領域に記録した後に、その記憶領域の場所を記録する領域である。

【００１８】図５はモジュール管理手段６４の動作を表わすフローチャート図である。

【００１９】図６はイベント管理手段６３の動作を表わすフローチャート図である。

【００２０】図７はスクリプト管理手段６２の動作を表わすフローチャート図である。

【００２１】図８は描画手段６１の動作を表わすフローチャート図である。

【００２２】以上のように構成された本実施例のマルチウインドウ制御方式について、以下にその動作を説明する。

【００２３】まず、モジュール管理手段６４では、リクエスト判定６４１でクライアント５からリクエストが送られてきているかを調べ、リクエストが送られてきていない場合は、入力判定６４８を実行する。リクエストが送られてきている場合はリクエスト保存処理６４２でリクエストをリクエスト保存領域６５に記録する。なお、リクエストは、リクエストを送ったクライアント５の識別記号、リクエストの識別記号、及びそれぞれのリクエストに固有の付加情報で構成されているものとする。リクエストを記録した後、イベント登録要求判定６４３、またはスクリプト登録要求判定６４５でリクエストの種類を調べる。

【００２４】リクエストがイベントの登録を要求するものである場合、イベント管理処理６４４によりイベント管理手段６３に実行が移る。なお、このリクエストには、入力装置４の操作等によって生成しなければならないイベントの識別記号、イベントの対応処理をクライアント５側で行なうか、サーバ６側で行なうかを示す記号、及びイベントの対応処理をサーバ６側で行なう場合に実行するスクリプトの識別記号が付加されているものとする。イベント管理手段６３では、まずリクエスト判定６３４でリクエストの有無を調べる。この場合リクエストが有るので、イベント登録６３５によりコンピュータの記憶領域に図３のようなイベント制御情報を作成し、その領域のクライアント部６３１にリクエストに付加されているクライアント５の識別記号を記録し、イベント部６３２にリクエストに付加されているイベントの識別記号を記録する。イベントへの対応処理をクライアント５側で行なうことを要求するリクエストである場合、スクリプト部６３３には何も記録しない。イベントへの対応処理をサーバ６側で行なうことを要求するリクエストである場合、スクリプト部６３３にはリクエストに付加されているスクリプトの識別記号を記録する。以上のようにイベントを登録した後、イベント管理手段６３の処理を終了し、モジュール管理手段６４の入力判定６４８に実行が移る。

【００２５】リクエストがスクリプトの登録を要求するものである場合、スクリプト管理処理６４６によりスクリプト管理手段６２に実行が移る。なお、このリクエストには登録するスクリプト、及びそのスクリプトの識別記号が付加されているものとする。スクリプト管理手段６２では、まずリクエスト判定６２４でリクエストの有無を調べる。この場合リクエストが有るので、スクリプト登録６２５により、コンピュータの記憶領域にスクリプトを記録する。更に別の記憶領域に図４のようなスクリプト制御情報を作成し、その領域のクライアント部６２１にリクエストに付加されているクライアント５の識別記号を記録し、スクリプト部６２２にリクエストに付加されているスクリプトの識別記号を記録し、保存部６２３にスクリプトを記録した記憶領域の場所を記録する。以上のようにスクリプトを登録した後、スクリプト管理手段６２の処理を終了し、モジュール管理手段６４の入力判定６４８に実行が移る。

【００２６】リクエストがイベントの登録を要求するものではなく、スクリプトの登録を要求するものでもない場合、描画処理６４７により描画手段６１に実行が移る。描画手段６１では、様々なリクエストの種類を判別して、リクエストに応じた処理を行なう。例えば、文字描画判定６１１で文字描画を要求するリクエストであるかどうかを判定し、文字描画を要求するリクエストである場合は、文字描画処理６１２で出力装置３に文字を表示させ、再びモジュール管理手段６４の入力判定６４８に実行を移す。

【００２７】モジュール管理手段６４はサーバ６を実行中、定期的に入力判定６４８で、コンピュータの利用者による入力装置４の操作の有無を判定する。入力装置４の操作が行なわれていない場合は、イベントを生成する必要がないため、リクエスト判定６４１に戻る。入力装置４の操作が行なわれた場合は、終了操作判定６４９で操作がサーバ６の終了操作であるかを判定し、終了操作であるならばサーバ６の実行を終了する。終了操作でな

い場合、イベント管理処理６４４によりイベント管理手段６３に実行が移る。

【００２８】イベント管理手段６３では、まずリクエスト判定６３４でリクエストの有無を調べる。既にリクエストの処理を終えているため、次のイベント生成６３６で、入力装置４の操作内容などからイベントを生成し、操作内容に応じてイベントの識別記号を割り付け、イベント保存領域６６に記録する。次にイベント登録判定６３７でイベントの登録の有無を判定する。入力装置４の操作の対象となるクライアント５用に作成したイベント制御情報があり、且つイベント部６３２に、生成したイベントの識別記号と同じ記号が記録されていれば、イベントが登録されているものとして、処理判定６３８を実行する。イベント制御情報が無いか、またはイベント制御情報のイベント部６３２に発生したイベントの識別記号と同じ記号が記録されていなければ、イベントに対応する処理を行なう必要がないものとして、何もせずにイベント管理手段６３の処理を終了し、モジュール管理手段６４のイベント転送判定６４１０に実行が移る。処理判定６３８ではイベントに対応する処理をサーバ６、またはクライアント５のどちらで行なうかを判定する。イベント制御情報のスクリプト部６３３に何も記録されていない場合は、イベントに対応する処理をクライアント６側で実行するものとして、イベント転送要求６３９でイベントをクライアント６に転送する様にモジュール管理手段６４に要求し、イベント管理手段６３の処理を終了し、モジュール管理手段６４のイベント転送判定６４１０に実行が移る。イベント制御情報のスクリプト部６３３にスクリプトの識別記号が記録されている場合は、イベントに対応する処理をサーバ６側で実行するものとして、スクリプト実行要求６３１０で、その識別記号で示されるスクリプトの実行要求をスクリプト管理手段６２に要求し、イベント管理手段６３の処理を終了し、モジュール管理手段６４のイベント転送判定６４１０に実行が移る。

【００２９】モジュール管理手段６４はイベント転送判定６４１０で、クライアント６へのイベント転送要求発生の有無を判定する。イベント転送要求が発生している場合、イベント転送処理６４１１でイベント保存領域６６に記録されたイベントを読みだし、クライアント５へ転送し、リクエスト判定６４１の処理に戻る。

【００３０】イベント転送要求が発生していない場合、スクリプト実行判定６４１２で、イベント管理手段６３からスクリプト管理手段６２へ、スクリプト実行要求が発生しているかどうかを判定する。スクリプト実行要求が発生していない場合、リクエスト判定６４１の処理に戻る。スクリプト実行要求が発生している場合、スクリプト管理処理６４６によりスクリプト管理手段６２に実行が移る。

【００３１】スクリプト管理手段６２では、まずリクエスト判定６２４でリクエストの有無を調べる。既にリクエストの処理を終えているため、次のスクリプト実行６２６でイベント管理手段６３から要求のあったスクリプトを実行する。スクリプト中に描画手段６１で実行するリクエストが含まれている場合は、そのリクエストをリクエスト保存領域６５に記録し、描画手段６１に描画処理要求を出し、スクリプト管理手段６２の処理を終了し、モジュール管理手段６４の描画処理判定６４１３に実行が移る。

【００３２】モジュール管理手段６４は描画処理判定６４１３で、スクリプト管理手段６２から描画手段６１へ描画処理要求が発生しているかどうかをを判定する。描画処理要求が発生していない場合、リクエスト判定６４１の処理に戻る。描画処理要求が発生している場合、描画処理６４７により描画手段６１に実行が移る。

【００３３】描画手段６１ではリクエスト保存領域６５に記録されているリクエストに応じた処理を行なった後、描画手段６１の処理を終了し、モジュール管理手段６４のリクエスト判定６４１に実行が移る。

【００３４】

【発明の効果】以上説明したように、本発明はコンピュータの利用者が頻繁に入力装置を操作する場合でも、クライアントとサーバ間でやり取りするイベント、及びリクエストの量が少ないため、これらをやり取りするための処理時間が大きくならず、クライアント、およびサーバの実行速度が低下しない。さらにクライアントとサーバが別々のコンピュータで動作している場合、ネットワークに流れるデータ量が増えることがなく、同じネットワークを利用している他のコンピュータの通信速度を低下させることがない。

【００３５】また、本発明は従来ならサーバ側で行なうイベントの対応処理を、クライアントの提供するスクリプトで行なうため、サーバのプログラムが巨大にならず、サーバの開発コストが安くなり、保守も容易となる。

【図面の簡単な説明】

【図１】本発明の一実施例を示す構成図である。

【図２】本発明の一実施例及び従来例を示す構成図である。

【図３】イベント制御情報を示す図である。

【図４】スクリプト制御情報を示す図である。

【図５】モジュール管理手段の動作を示すフローチャート図である。

【図６】イベント管理手段の動作を示すフローチャート図である。

【図７】スクリプト管理手段の動作を示すフローチャート図である。

【図８】描画手段の動作を示すフローチャート図である。

【符号の説明】

9

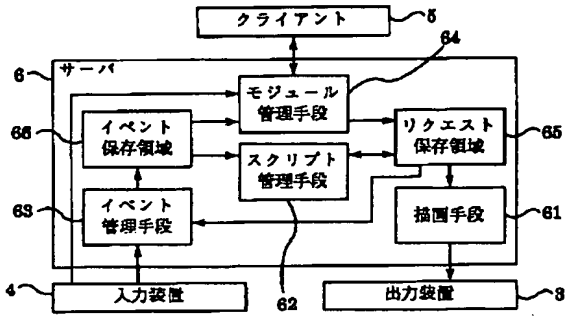
1…コンピュータ、
2…ネットワーク、
3…出力装置、
4…入力装置、
5…クライアント、
6…サーバ、
６１…描画手段、
６２…スクリプト管理手段、
６３…イベント管理手段、

10

６４…モジュール管理手段、
６５…リクエスト保存領域、
６６…イベント保存領域、
６２１…クライアント部、
６２２…スクリプト部、
６２３…保存部、
６３１…クライアント部、
６３２…イベント部、
６３３…スクリプト部。

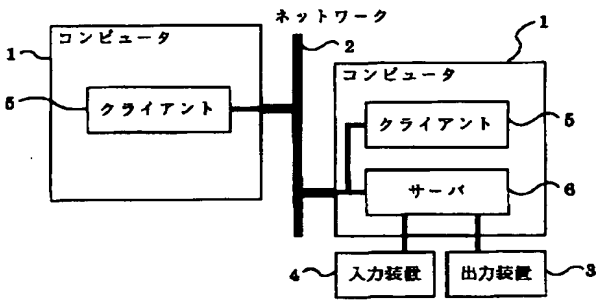
【図１】

図 1



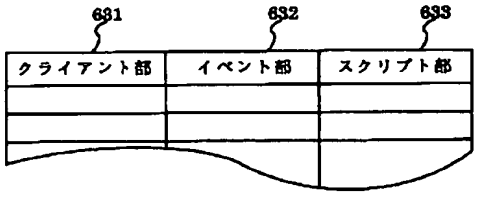
【図２】

図 2



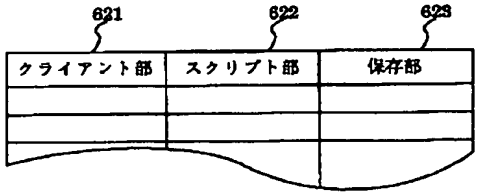
【図３】

図 3



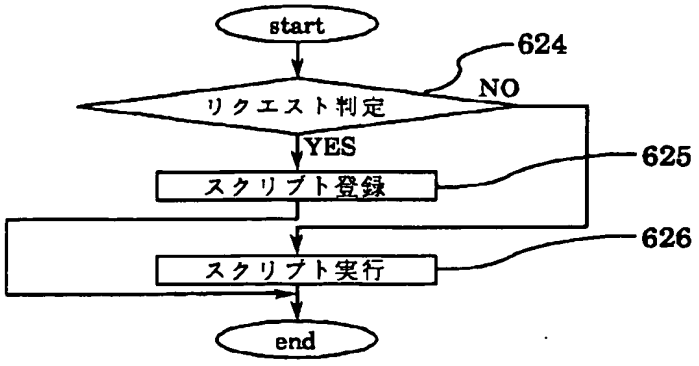
【図４】

図 4



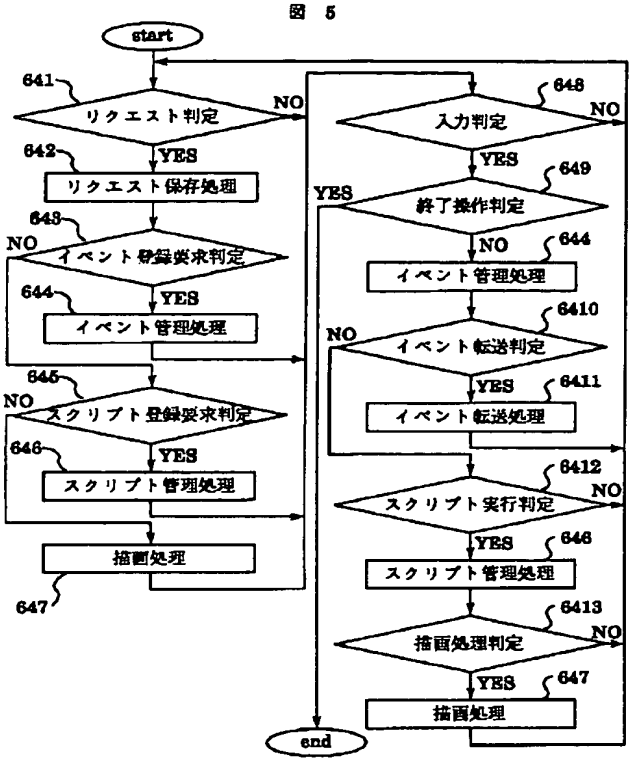
【図７】

図 7



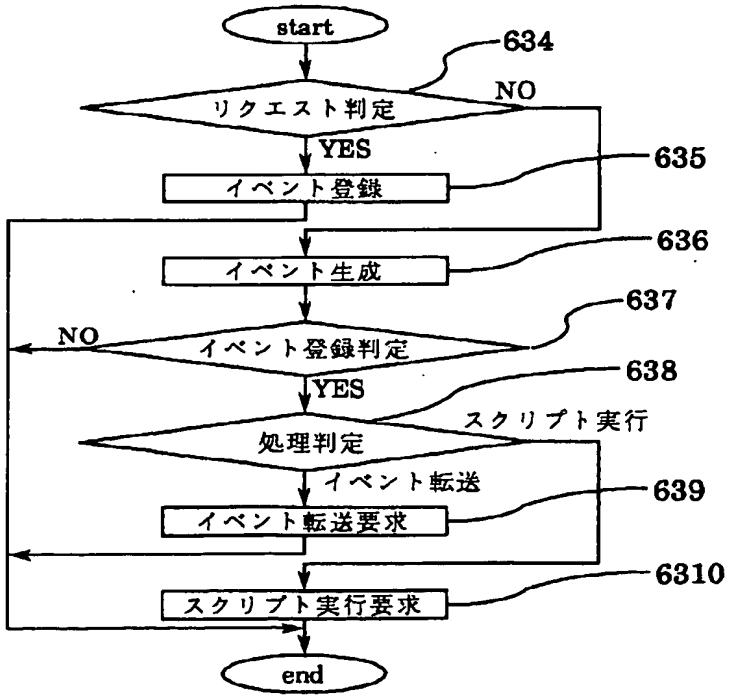
【図５】

図 5


start
641
リクエスト判定
NO
YES
642
リクエスト保存処理
643
イベント登録要求判定
NO
YES
644
イベント管理処理
645
スクリプト登録要求判定
NO
YES
646
スクリプト管理処理
描画処理
647
648
入力判定
NO
YES
649
終了操作判定
YES
NO
644
イベント管理処理
6410
イベント転送判定
NO
YES
6411
イベント転送処理
6412
スクリプト実行判定
NO
YES
646
スクリプト管理処理
6413
描画処理判定
NO
YES
647
描画処理
end

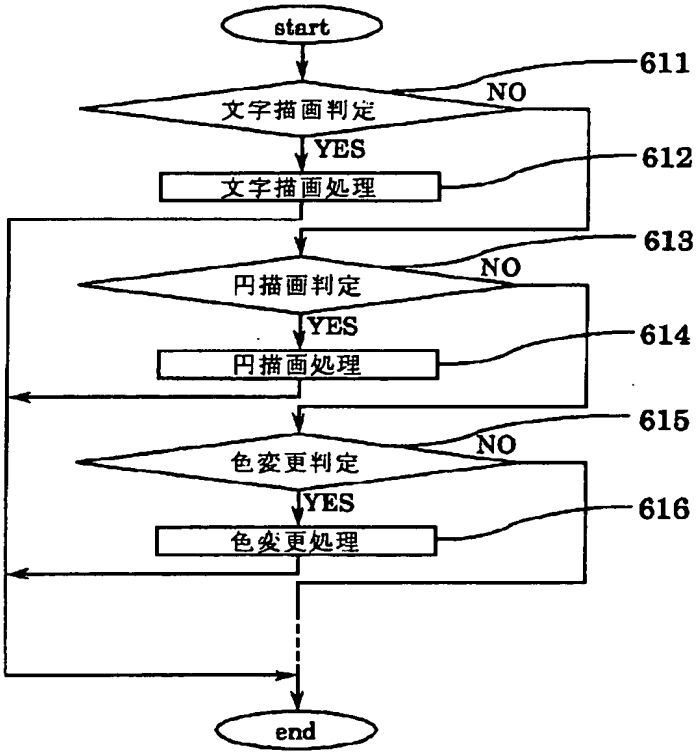
【図６】

図　6


start
634
リクエスト判定
NO
YES
635
イベント登録
636
イベント生成
637
NO
イベント登録判定
YES
638
スクリプト実行
処理判定
イベント転送
639
イベント転送要求
6310
スクリプト実行要求
end

【図８】

図　8



フロントページの続き

(72)発明者　羽田　功一
神奈川県横浜市戸塚区吉田町292番地株式会社日立マイクロソフトウェアシステムズ内

(72)発明者　鈴木　一成
神奈川県横浜市戸塚区吉田町292番地株式会社日立マイクロソフトウェアシステムズ内